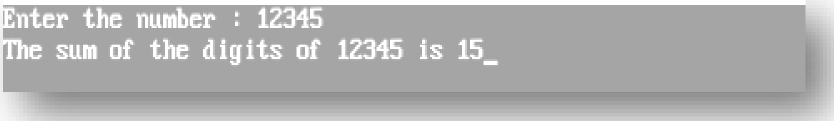
Computer Lab Practical Question Bank
B.Com (Computer Applications) CBCS III Year, Semester – V
Subject: Object Oriented Programming in C++

1. Write a program to print the sum of digits of a given number?

```
#include<iostream.h>
#include<conio.h>
int main()
{
   int val, num, sum = 0;
    cout << "Enter the number : ";
   cin >> val;
   num = val;
   while (num != 0)
   {   sum = sum + num % 10;
      num = num / 10;
   }
   cout << "The sum of the digits of "<< val << " is " << sum;
}
```

Output:

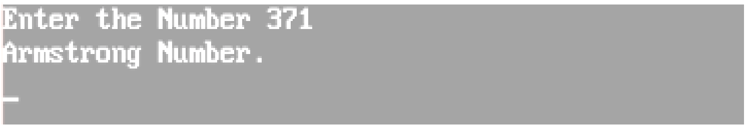


```
Enter the number : 12345
The sum of the digits of 12345 is 15_
```

2. Write a program to check whether the given number is Armstrong or not ?

```
#include <iostream.h>
#include <conio.h>
int main()
{
int n,r,sum=0,temp;
cout<<"Enter the Number=  ";
cin>>n;
temp=n;
while(n>0)
{         r=n%10;
          sum=sum+(r*r*r);
          n=n/10;
}
if(temp==sum)
cout<<"Armstrong Number."<<endl;
else
cout<<"Not Armstrong Number."<<endl;
return 0;
}
```

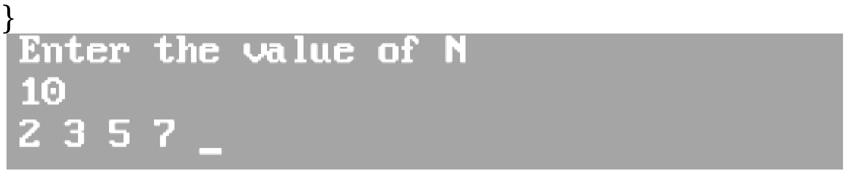
Output:



```
Enter the Number 371
Armstrong Number.
_
```

3. Write a Program to print prime numbers from 2 to n where n is a natural number given ?

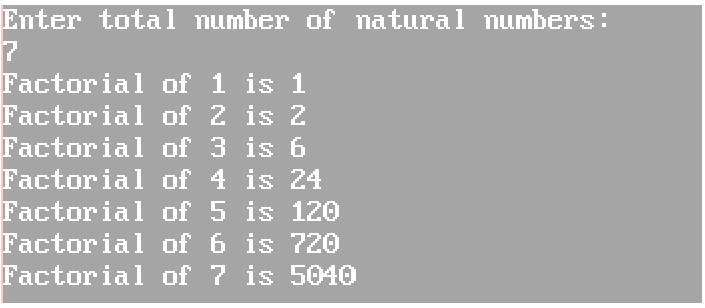
```
#include<iostream.h>
#include<conio.h>
 int main()
{
   int N, i, j, isPrime, n;
   cout << "Enter the value of N\n";
   cin >> N;
for(i = 2; i <= N; i++)
{
       isPrime = 0;
       for(j = 2; j <= i/2; j++)
      {
           if(i % j == 0)
           {
              isPrime = 1;
              break;
       }
}
if(isPrime==0 && N!= 1)
           cout << i << " ";
}
return 0;
}
```



```
Enter the value of N
10
2 3 5 7 _
```

4.Write a Program to show the use of inline function.

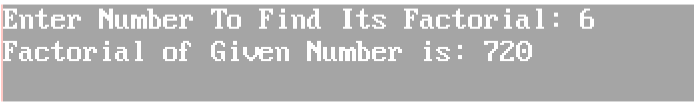
```
#include <iostream.h>
#include <conio.h>
inline void factorial(int x)
{
   int i,f=1,p=1;
   for(i=1;i<=x;i++)
   {      f=f*i;   }
   cout<<"Factorial of "<<x<<" is "<<f<<endl;
}
int main()
{
   int i, N;
   cout<<"Enter total number of natural numbers:"<<endl;
   cin>>N;
   for(i=1;i<=N;i++)
   {       factorial(i);       }
   getch();
   return 0;
}
```



```
Enter total number of natural numbers:
7
Factorial of 1 is 1
Factorial of 2 is 2
Factorial of 3 is 6
Factorial of 4 is 24
Factorial of 5 is 120
Factorial of 6 is 720
Factorial of 7 is 5040
```

5. Write a Program to find the factorial of a given number.

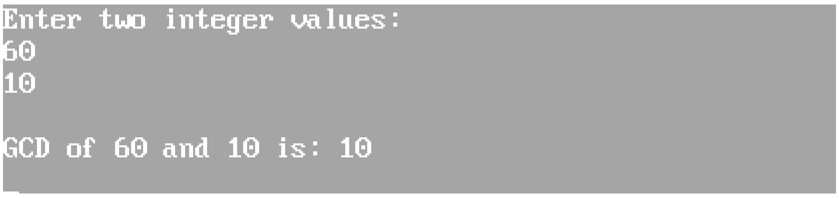
```
#include <iostream.h>
int main()
{
int number,factorial=1;
cout << "Enter Number To Find Its Factorial: ";
cin>>number;
  for(int i=1;i<=number;i++)
  {
    factorial=factorial*i;
  }
  cout<<"Factorial of Given Number is: " << factorial << endl;
  return 0;
}
```



```
Enter Number To Find Its Factorial: 6
Factorial of Given Number is: 720
```

6. Write a Program to calculate the GCD of two numbers

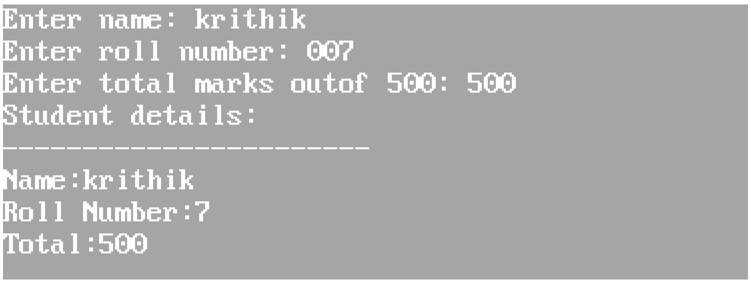
```
#include <iostream.h>
int main()
{
  int x, y, i, gcd;
  cout << "Enter two integer values:" << endl;
  cin >> x >> y;
  for (i = 1; i <= x && i <= y; i++)
  {
    if (x % i == 0 && y % i == 0)
      gcd = i;
  }
  cout << "\nGCD of " << x << " and " << y << " is: " << gcd << endl;
  return 0;
}
```



```
Enter two integer values:
60
10

GCD of 60 and 10 is: 10
```

7. Write a Program to read the student name, roll number ,marks and display the same using class and object

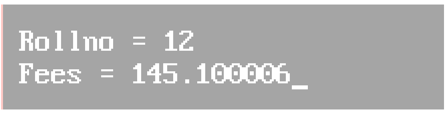
```
#include <iostream.h>
class student
{
        private:
                char  name[30];
                int   rollNo;
                int   total;
        public:
                //member function to get student's details
                void getDetails(void);
                //member function to print student's details
                void putDetails(void);
};
//member function definition, outside of the class
void student::getDetails(void){
        cout << "Enter name: " ;
        cin >> name;
        cout << "Enter roll number: ";
        cin >> rollNo;
        cout << "Enter total marks outof 500: ";
        cin >> total;
}
//member function definition, outside of the class
void student::putDetails(void){
        cout << "Student details:\n";
        cout << "Name:"<< name << ",Roll Number:" << rollNo << ",Total:" << total ;
}
int main()
{
        student std;            //object creation
        std.getDetails();
        std.putDetails();
        return 0;
}
```



```
Enter name: krithik
Enter roll number: 007
Enter total marks outof 500: 500
Student details:
-----------------------
Name:krithik
Roll Number:7
Total:500
```

8. Write a Program to show the usage of private member function

```
# include <iostream.h>
# include <conio.h>
class student
{
     private:
     int rn;
     float fees;
     void read()
     {
          rn=12;
          fees=145.10;
    }
   public:
     void show()
    {
        read();
        cout<<"\n Rollno = "<<rn;
        cout<<"\n Fees = "<<fees;
    }
  };
  void main ( )
  {
            clrscr ( );
            student st;
            // st.read ( );  // not accessible
            st.show ( );
            getch();
     }
```

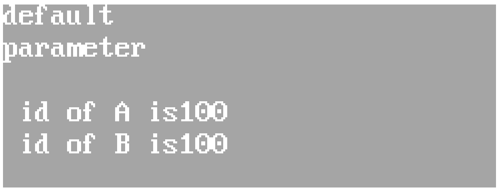


```
Rollno = 12
Fees = 145.100006_
```

9. Write a Program to implement default constructor ,parameterized constructor and copy constructor

```
#include<iostream.h>
#include<conio.h>
class code
{
   int id;
   public:
   code()                //default constructor
   {
      cout<<"default";
   }
   code(int a)          //parameterized constructor
   {
   cout<<"parameter";
```

```
    id=a;
    }
    code(code &x)     //copy constructor
    {
      id=x.id;          //copy in the value
    }
    void display(void)
    {
       cout<<id;
    }
};
void main()
{
    clrscr();
    code A(100);    //object A is created and initialized
    code B(A);      //copy constructor called
    cout<<"\n id of A is";A.display();
    cout<<"\n id of B is";B.display();
    getch();
}
```

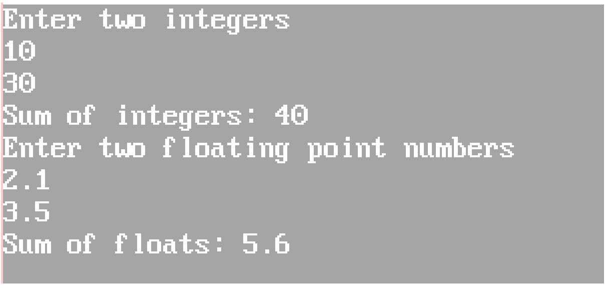


```
default
parameter

 id of A is100
 id of B is100
```

10.Write a Program to implement function overloading.

```
#include <iostream.h>
 long add(long, long);
float add(float, float);
 int main()
{
   long a, b, x;
   float c, d, y;
   cout << "Enter two integers\n";
   cin >> a >> b;
   x = add(a, b);
   cout << "Sum of integers: " << x << endl;
   cout << "Enter two floating point numbers\n";
   cin >> c >> d;
     y = add(c, d);
     cout << "Sum of floats: " << y << endl;
     return 0;
}
long add(long x, long y)
{          long sum;
           sum = x + y;
           return sum;
}
```

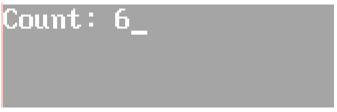
```
 float add(float x, float y)
{
  float sum;
  sum = x + y;
  return sum;
}
```



```
Enter two integers
10
30
Sum of integers: 40
Enter two floating point numbers
2.1
3.5
Sum of floats: 5.6
```

11.Write a Program to implement operator overloading.

```
#include <iostream.h>
class Test
{
  private:
     int count;
  public:
      Test(): count(5){}
      void operator ++()
      {
         count = count+1;
        }
      void Display() { cout<<"Count: "<<count; }
};
void  main()
{
   Test t;
   ++t;
   t.Display();
   }
```

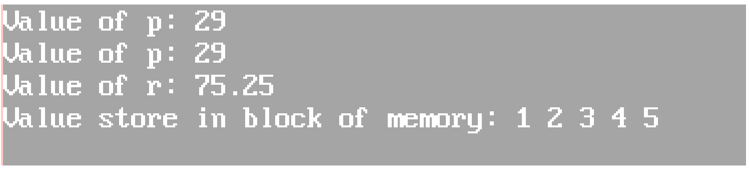


```
Count: 6_
```

12.Write a Program to implement dynamic memory allocation and de allocation using new and delete operators using class and object

```
#include <iostream.h>
 int main ()
{
int* p = NULL;
p = new int;
   if (!p)
   cout << "allocation of memory failed\n";
   else
   {
       *p = 29;
```

```
            cout << "Value of p: " << *p << endl;
          }
        float *r = new float(75.25);
        cout << "Value of r: " << *r << endl;
        int n = 5;
        int *q = new int[n];
         if (!q)
        cout << "allocation of memory failed\n";
          else
          {
            for (int i = 0; i < n; i++)
            q[i] = i+1;
            cout << "Value store in block of memory: ";
            for (int i = 0; i < n; i++)
            cout << q[i] << " ";
          }
         delete p;
         delete r;
         delete[] q;
         return 0;
        }
```

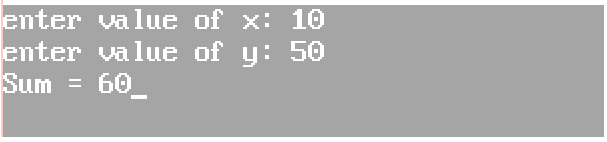


```
Value of p: 29
Value of p: 29
Value of r: 75.25
Value store in block of memory: 1 2 3 4 5
```

13.Write a Program to demonstrate multiple inheritance

```
#include<iostream.h>
class A
{
        public:
        int x;
        void getx()
        {       cout << "enter value of x: ";
                cin >> x;
        }
};
class B
{
        public:
        int y;
        void gety()
        {
          cout << "enter value of y: "; cin >> y;
        }
};
```

```
class C : public A, public B
{
        public:
        void sum()
        {
           cout << "Sum = " << x + y;
        }
};
int main()
{
         C obj1;
         obj1.getx();
         obj1.gety();
         obj1.sum();
         return 0;
}
```

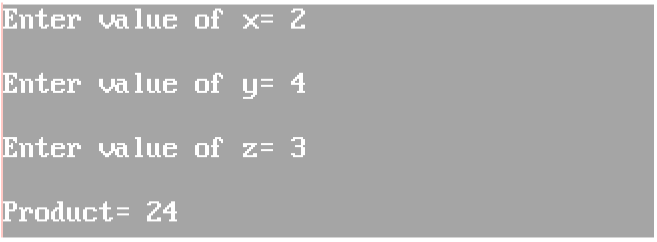


```
enter value of x: 10
enter value of y: 50
Sum = 60_
```

14.Write a Program to demonstrate multilevel inheritance

```
#include <iostream.h>
class base
{
        public:
        int x;
        void getdata()
        {
        cout << "Enter value of x= "; cin >> x;
        }
};
class derive1 : public base
{
        public:
        int y;
        void readdata()
        {
           cout << "\nEnter value of y= "; cin >> y;
        }
};
class derive2 : public derive1
{
        private:
        int z;
        public:
        void indata()
        {
```

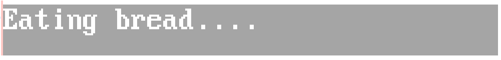
```
        cout << "\nEnter value of z= ";
        cin >> z;
        }
        void product()
        {
           cout << "\nProduct= " << x * y * z;
        }
};
int main()
{
    derive2 a;
    a.getdata();
    a.readdata();
    a.indata();
    a.product();
    return 0;
}
```



```
Enter value of x= 2

Enter value of y= 4

Enter value of z= 3

Product= 24
```

15. Write a Program to demonstrate dynamic polymorphism using method overriding and dynamic method dispatch.

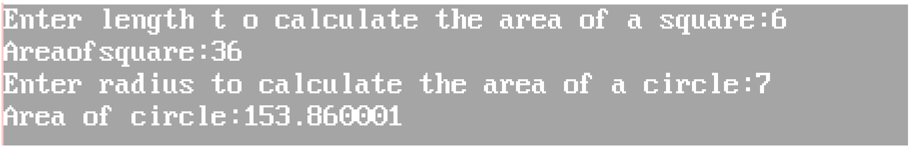
```
#include<iostream.h>
Class animal
{
      Public:
                Void eat( )
                {         cout<<”Eating.....”;    }
};
Class Dog : Public animal
{
      Public:
                Void eat( )
                {         cout<<”Eating Bread.....”;     }
};
int main ( )
{
animal   *a;
Dog d;
a=&d;
a->eat( );
return 0;
}
```



```
Eating bread....
```

16.Write a Program to implement polymorphism using pure virtual function

```
#include<iostrearm.h>
Class Shape
{
Protected:
float a;
public:
void getDataQ
{
cin>>a;
}
Virtual float calculateArea()=0;
};
Class Square: public Shape
{ public:
float calculateArea()
{   return a*a;   }
};
Class Circle: public Shape
{   public:
float calculateArea()
{   return 3.14*a*a;     }
};
int main( )
{
Square s;
Circle c;
cout<<"Enterlengthtocalculatetheareaofasquare:";
s.getData();
cout<<"Areaofsquare:"<<s.calculateArea( );
cout<<"\nEnterradiustocalculatetheareaofa circle:";
c.getData();
cout<<"Area of circle:”<<c.calcuiateArea();
return 0;
}
```

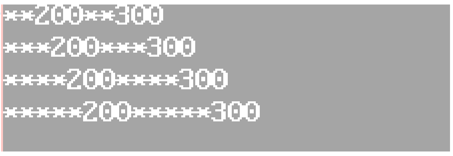


```
Enter length t o calculate the area of a square:6
Areaofsquare:36
Enter radius to calculate the area of a circle:7
Area of circle:153.860001
```

17.Write a Program to show usage of setw() and setfill()

```
#include <iostream.h>
#include <iomanip.h>
```

```
void main ( void)
{
   int   a,b;
   a = 200;
   b = 300;
   cout << setfill ( ' * ' );
   cout << setw (5) << a << setw (5) << b << endl;
   cout << setw (6) << a << setw (6) << b << endl;
   cout << setw (7) << a << setw (7) << b << endl;
   cout << setw (8) << a << setw (8) << b << endl;
}
```

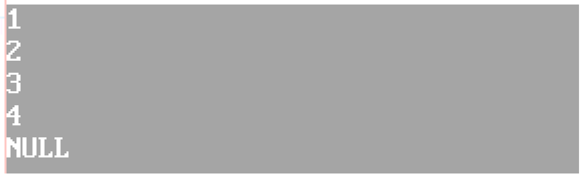


```
**200**300
***200***300
****200****300
*****200*****300
```

18. Write a Program to create and insert elements in a linked list

```
#include <iostream.h>
struct node
{
   int data;
   node *next;
};
class linked_list
{
private:
   node *head,*tail;
public:
   linked_list()
   {
      head = NULL;
      tail = NULL;
   }
   void add_node(int n)
   {
      node *tmp = new node;
      tmp->data = n;
      tmp->next = NULL;
      if(head == NULL)
      {
         head = tmp;
         tail = tmp;
      }
else
      {
         tail->next = tmp;
         tail = tail->next;
      }
```

```
  }
  node* gethead()
  {
    return head;
  }
  static void display(node *head)
  {
    if(head == NULL)
    {
      cout << "NULL" << endl;
    }
    else
    {
      cout << head->data << endl;
      display(head->next);
    }
  }
  static void concatenate(node *a,node *b)
  {
    if( a != NULL && b!= NULL )
    {
      if (a->next == NULL)
        a->next = b;
      else
        concatenate(a->next,b);
    }
    else
    {
      cout << "Either a or b is NULL\n";
    }
  }
};
int main()
{
  linked_list a;
  a.add_node(1);
  a.add_node(2);
  linked_list b;
  b.add_node(3);
  b.add_node(4);
  linked_list::concatenate(a.gethead(),b.gethead());
  linked_list::display(a.gethead());
  return 0;
}
```

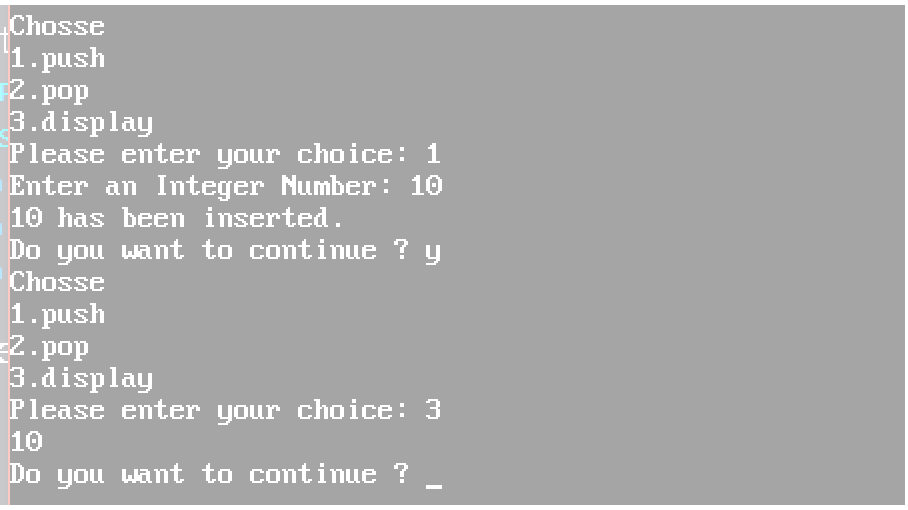


```
1
2
3
4
NULL
```

19.Write a Program for implementation of stack using array

```
#include<iostream>
#define MAX 5
 using namespace std;
 int STACK[MAX],TOP;
 void initStack()
{
   TOP=-1;
}
int isEmpty(){
   if(TOP==-1)
      return 1;
   else
      return 0;
}
int isFull(){
   if(TOP==MAX-1)
      return 1;

   else
      return 0;
}
void push(int num){
   if(isFull()){
      cout<<"STACK is FULL."<<endl;
      return;
   }
   ++TOP;
   STACK[TOP]=num;
   cout<<num<<" has been inserted."<<endl;
}
void display(){
   int i;
   if(isEmpty()){
      cout<<"STACK is EMPTY."<<endl;
      return;
   }
   for(i=TOP;i>=0;i--){
      cout<<STACK[i]<<" ";
   }
   cout<<endl;
}
```

```
void pop(){
  int temp;
  if(isEmpty()){
    cout<<"STACK is EMPTY."<<endl;
    return;
  }
  temp=STACK[TOP];
  TOP--;
  cout<<temp<<" has been deleted."<<endl;
}
int main(){
  int num;
  initStack();
  char ch;
  do{
      int a;
      cout<<"Chosse \n1.push\n"<<"2.pop\n"<<"3.display\n";
      cout<<"Please enter your choice: ";
      cin>>a;
      switch(a)
      {
        case 1:
          cout<<"Enter an Integer Number: ";
          cin>>num;
          push(num);
        break;
        case 2:
          pop();
          break;
        case 3:
          display();
          break;
        default :
        cout<<"An Invalid Choice!!!\n";
      }
      cout<<"Do you want to continue ? ";
      cin>>ch;
      }while(ch=='Y'||ch=='y');
  return 0;
}
```

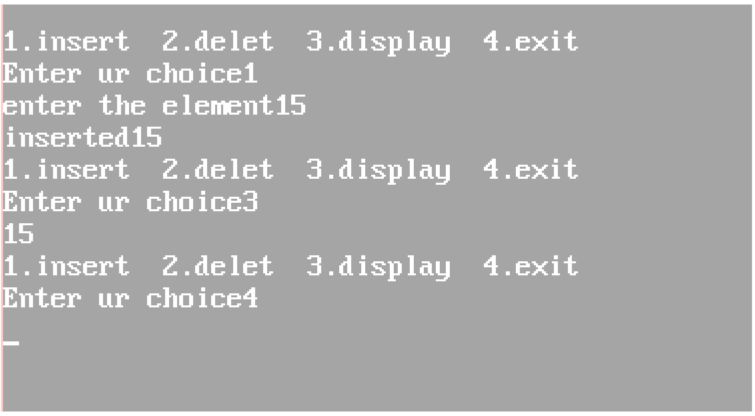


```
Chosse
1.push
2.pop
3.display
Please enter your choice: 1
Enter an Integer Number: 10
10 has been inserted.
Do you want to continue ? y
Chosse
1.push
2.pop
3.display
Please enter your choice: 3
10
Do you want to continue ? _
```

20.Write a Program for implementation of queue using array

```
#include<iostream>
#include<conio.h>
#include<stdlib.h>
class queue
{
        int queue1[5];
        int rear,front;
    public:
        queue()
         {
            rear=-1;
            front=-1;
         }
        void insert(int x)
        {
           if(rear >  4)
            {
              cout <<"queue over flow";
              front=rear=-1;
              return;
            }
            queue1[++rear]=x;
            cout <<"inserted" <<x;
        }
        void delet()
        {
           if(front==rear)
             {
               cout <<"queue under flow";
               return;
             }
             cout <<"deleted" <<queue1[++front];
         }
        void display()
```

```
          {
            if(rear==front)
             {
                cout <<" queue empty";
                return;
             }
            for(int i=front+1;i<=rear;i++)
            cout <<queue1[i]<<" ";
          }
   };

   main()
   {
      int ch;
      queue qu;
      while(1)
       {
          cout <<"\n1.insert  2.delet  3.display  4.exit\nEnter ur choice";
          cin >> ch;
          switch(ch)
           {
            case 1:   cout <<"enter the element";
                      cin >> ch;
                      qu.insert(ch);
                      break;
            case 2:  qu.delet();  break;
            case 3:  qu.display();break;
            case 4: exit(0);
            }
        }
   return (0);
   }
```



```
1.insert  2.delet  3.display  4.exit
Enter ur choice1
enter the element15
inserted15
1.insert  2.delet  3.display  4.exit
Enter ur choice3
15
1.insert  2.delet  3.display  4.exit
Enter ur choice4
_
```